Integrating Specialized Procedures in Proof Systems

Vishal Sikka
Computer Science Department
Stanford University
Stanford, CA - 94305
vishal@cs.stanford.edu

Introduction

We present the outline of a simple but powerful scheme for describing procedures that can be used by an automatic theorem prover. Our approach is to *describe* specialized procedures to a theorem prover by adding procedure description axioms to its set of facts, instead of *building in* these procedures by using attachments. Our work can be viewed as an extension of the hybrid reasoning techniques based on attachments (Myers91).

In this abstract we briefly describe our approach and state some of its advantages. In (Sikka94), we present this work in detail and formally show how the full expressibility of attachment-like approaches can be achieved in a simple logic-theoretic way. The work described here is part of the author's Ph.D. thesis research in collaboration with Prof. Michael Genesereth.

A Simple Integration Scheme

We treat a programming environment as a 3-tuple $\Im : \langle \mathcal{P}, \mathcal{D}, \mathcal{E} \rangle$, of procedures ($\mathcal{P}$), data structures ($\mathcal{D}$), and an evaluation function $\mathcal{E} : \mathcal{P} \times \mathcal{D}^* \rightarrow \mathcal{D}$. We integrate procedures from such an environment into a first order deductive calculus. Here we briefly describe its syntax and semantics. The language $\mathcal{L}_{\mathcal{M}}$ is a construction from a standard first order language $\mathcal{L}$, a set P_N of procedure names (of $\mathcal{P}$ in $\Im$), and an n-ary function symbol *apply*.

We can use $\mathcal{L}_{\mathcal{M}}$ to describe procedures in a programming environment and their relationship to functions and relations. For example, the sentence
$\forall x.\forall y.\ plus(x,y) = apply(+,x,y)$
relates the procedure + to the function *plus*.

Every expression containing the function symbol *apply* has the following semantics:
$apply(p, t_1, t_2, \ldots, t_n)^{\Im} = \mathcal{E}(\phi, \tau_1, \tau_2, \ldots, \tau_n)$
where $p^{\Im} = \phi$, and $t_i{}^{\Im} = \tau_i$ for every i, $1 \leq i \leq n$. Every other expression in $\mathcal{L}_{\mathcal{M}}$ has standard first order semantics. We add to the deductive machinery an inference rule that allows us to replace a ground term containing the function symbol *apply* with the result of applying the associated procedure on the corresponding argument terms. This framework allows us to prove the following result. It is formally described and proved in (Sikka94).

Theorem 1 *For every set of attachments, there is a set of sentences in $\mathcal{L}_{\mathcal{M}}$ that entail the same deductions.*

Summary

There are four principal advantages to using this approach for integration:

1. Attachments are principally *substitutional* in nature. Our scheme can be used to describe and use specialized procedures that perform more complex types of reasoning.
2. Attachment-based approaches are limited in the kinds of conditions that can be imposed on the invocation of procedures. With our scheme the language of representation, i.e. $\mathcal{L}_{\mathcal{M}}$ itself can be used to prescribe any conditions on the use of the attached procedures.
3. Sentences with the function symbol *apply* can be used to reason about the attached procedures. Since sentences containing the function symbol *apply* are ordinary sentences, composition axioms for attached procedures can be described and reasoned with ordinarily.
4. Describing attached procedures to a theorem prover is much easier, from an implementational viewpoint, than building in the attachment for every function and relation symbol that has an associated procedure.

References

Myers, K.L. 1991, Universal Attachments: A Logical Framework for Hybrid Reasoning, Ph.D. Thesis, Dept. of Computer Science, Stanford University, 1991.

Sikka, V.I., and Genesereth, M.R. 1994. Integrating Specialized Procedures in Proof Systems. Logic Group Technical Report LOGIC-94-3, Dept. of Computer Science, Stanford University. Also to appear in the Workshop on Metatheoretic Extensibility of Automated Reasoning Systems, *CADE-12*, Nancy, France.